Modify System Image: Patch System Image, Sub-technique T1601.001 - Enterprise

Archived: 2026-04-05 17:38:44 UTC

Adversaries may modify the operating system of a network device to introduce new capabilities or weaken existing defenses.[1] [2] [3] [4] [5] Some network devices are built with a monolithic architecture, where the entire operating system and most of the functionality of the device is contained within a single file. Adversaries may change this file in storage, to be loaded in a future boot, or in memory during runtime.

To change the operating system in storage, the adversary will typically use the standard procedures available to device operators. This may involve downloading a new file via typical protocols used on network devices, such as TFTP, FTP, SCP, or a console connection. The original file may be overwritten, or a new file may be written alongside of it and the device reconfigured to boot to the compromised image.

To change the operating system in memory, the adversary typically can use one of two methods. In the first, the adversary would make use of native debug commands in the original, unaltered running operating system that allow them to directly modify the relevant memory addresses containing the running operating system. This method typically requires administrative level access to the device.

In the second method for changing the operating system in memory, the adversary would make use of the boot loader. The boot loader is the first piece of software that loads when the device starts that, in turn, will launch the operating system. Adversaries may use malicious code previously implanted in the boot loader, such as through the ROMMONkit method, to directly manipulate running operating system code in memory. This malicious code in the bootloader provides the capability of direct memory manipulation to the adversary, allowing them to patch the live operating system during runtime.

By modifying the instructions stored in the system image file, adversaries may either weaken existing defenses or provision new capabilities that the device did not have before. Examples of existing defenses that can be impeded include encryption, via Weaken Encryption, authentication, via Network Device Authentication, and perimeter defenses, via Network Boundary Bridging. Adding new capabilities for the adversary’s purpose include Keylogging, Multi-hop Proxy, and Port Knocking.

Adversaries may also compromise existing commands in the operating system to produce false output to mislead defenders. When this method is used in conjunction with Downgrade System Image, one example of a compromised system command may include changing the output of the command that shows the version of the currently running operating system. By patching the operating system, the adversary can change this command to instead display the original, higher revision number that they replaced through the system downgrade.

When the operating system is patched in storage, this can be achieved in either the resident storage (typically a form of flash memory, which is non-volatile) or via TFTP Boot.

When the technique is performed on the running operating system in memory and not on the stored copy, this technique will not survive across reboots. However, live memory modification of the operating system can be combined with ROMMONkit to achieve persistence.
